Research | Published: 26 June 2023

WebAssembly as an Enabler for Next Generation Serverless Computing

Vojdan Kjorveziroski & Sonja Filiposka

Journal of Grid Computing **21**, Article number: 34 (2023)

Abstract

WebAssembly is a new binary instruction format and runtime environment capable of executing both client side and server side workloads. With its numerous advantages, including drastically reduced cold start times, efficiency, easy portability, and compatibility with the most popular programming languages today, it has the potential to revolutionize serverless computing. We evaluate the impact of WebAssembly in terms of serverless

computing, building on top of existing research related to WebAssembmly in cloud and edge environments. To this end, we introduce a novel benchmarking suite comprised of 13 different functions, compatible with WebAssembly, and focusing on both microbenchmarking and real-world workloads. We also discuss possibilities of integrating WebAssembly runtimes with the application programming interfaces and command line interfaces of popular container runtimes, representing an initial step towards potential reuse of existing orchestration engines in the future, thus solving the open issue of WebAssembly workload scheduling. We evaluate the performance of such an integration by comparing the cold start delays and total execution times of three WebAssembly runtimes: WasmEdge, Wasmer, and Wasmtime to the performance of the containerd container runtime, using distroless and distro-oriented container images. Results show that WebAssembly runtimes show better results in 10 out of 13 tests, with Wasmtime being the fastest WebAssembly runtime among those evaluated. Container runtimes still offer better compute performance for complex workloads requiring larger execution times, in cases where cold start times are negligible compared to the total execution time.

Availability of data and material

The generated raw data, software, and outputs from the data analysis are made publicly available under a permissive license on https://github.com/korvoj/wasm-serverless-benchmarks

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing (2009)

2. Mell, P., Grance, T.: The NIST definition of cloud computing. Technical Report NIST Special Publication (SP) 800-145, National Institute of Standards and Technology (September 2011). https://doi.org/10.6028/NIST.SP.800-145

3. Duan, Y., Fu, G., Zhou, N., Sun, X., Narendra, N.C., Hu, B.: Everything as a service (XaaS) on the cloud: origins, current and future trends. In: 2015 IEEE 8th International Conference on Cloud Computing, pp. 621–628 (2015). https://doi.org/10.1109/CLOUD.2015.88

4. Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.-C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N., Gonzalez, J.E., Popa, R.A., Stoica, I., Patterson, D.A.: Cloud programming simplified: a Berkeley View on serverless computing. arXiv:1902.03383 [cs] (2019)

5. Kratzke, N.: A brief history of cloud application architectures. Applied Sciences **8**(8), 1368 (2018). https://doi.org/10.3390/app8081368

6. Wen, J., Liu, Y., Chen, Z., Chen, J., Ma, Y.: Characterizing commodity serverless computing platforms. Journal of Software: Evolution and Process **n/a**(n/a), 2394. https://doi.org/10.1002/smr.2394

7. El Ioini, N., Hästbacka, D., Pahl, C., Taibi, D.: Platforms for serverless at the edge: A Review. In: Zirpins, C., Paraskakis, I., Andrikopoulos, V., Kratzke, N., Pahl, C., El Ioini, N., Andreou, A.S., Feuerlicht, G., Lamersdorf, W., Ortiz, G., Van den Heuvel, W.-J., Soldani, J., Villari, M., Casale, G., Plebani, P. (eds.) Advances in service-oriented and cloud computing vol. 1360, pp. 29–40. Springer International Publishing. Cham (2021)

8. Li, J., Kulkarni, S.G., Ramakrishnan, K.K., Li, D.: Analyzing open-source serverless platforms: characteristics and performance. pp 15–20 arXiv:2106.03601 [cs] (2021) https://doi.org/10.18293/SEKE2021-129

9. Cloudflare Workers. https://workers.cloudflare.com/ Accessed 2022-11-09

10. Pfandzelter, T., Bermbach, D.: IoT data processing in the fog: Functions, streams, or batch processing? In: 2019 IEEE International Conference on Fog Computing (ICFC). pp. 201–206. IEEE Prague, Czech Republic (2019). https://doi.org/10.1109/ICFC.2019.00033

11. Varghese, B., Buyya, R.: Next generation cloud computing: New trends and research directions. Future Generation Computer Systems **79**, 849–861 (2018). https://doi.org/10.1016/j.future.2017.09.020

12. Salehe, M., Hu, Z., Mortazavi, S.H., Mohomed, I., Capes, T.: VideoPipe: Building Video stream processing pipelines at the edge. In: Proceedings of the 20th International Middleware Conference Industrial Track pp. 43–49 ACM Davis CA USA (2019). https://doi.org/10.1145/3366626.3368131

13. Kjorveziroski, V., Filiposka, S., Trajkovik, V.: IoT serverless computing at the edge: A Systematic Mapping Review. Computers **10**(10), 130 (2021). https://doi.org/10.3390/computers10100130

14. Hellerstein, J.M., Faleiro, J., Gonzalez, J.E., Schleier-Smith, J., Sreekanti, V., Tumanov, A., Wu, C.: Serverless computing:One step forward, two steps back. In:CIDR 2019,Monterey. CA (2018)

15. Kjorveziroski, V., Canto, C.B., Roig, P.J., Gilly, K., Mishev, A., Trajkovik, V., Filiposka, S.: IoT serverless computing at the edge: Open issues and research direction. Transactions on Networks and Communications vol.**9**(4), pp. 1–33 (2021). https://doi.org/10.14738/tnc.94.11231

16. Bocci, A., Forti, S., Ferrari, G.-L., Brogi, A.: Secure FaaS orchestration in the fog: How far are we? Computing **103**(5), 1025–1056 (2021). https://doi.org/10.1007/s00607-021-00924-y

17. Kjorveziroski, V., Filiposka, S.: Kubernetes distributions for the edge: Serverless performance evaluation. The Journal of Supercomputing **78**(11), 13728–13755 (2022). https://doi.org/10.1007/s11227-022-04430-6

18. Wang, B., Ali-Eldin, A., Shenoy, P.: LaSS: Running latency sensitive serverless computations at the edge. In:Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing, Association for Computing Machinery. New York, NY, USA, pp. 239–251 (2020)

19. Agarwal, S., Rodriguez, M.A., Buyya, R.: A reinforcement learning approach to reduce serverless function cold start frequency. In: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 797–803 https://doi.org/10.1109/CCGrid51090.2021.00097 (2021)

20. Murphy, S., Persaud, L., Martini, W., Bosshard, B.: On the use of web assembly in a serverless context.In: Paasivaara, M., Kruchten, P. (eds.) Agile Processes in Software Engineering and Extreme Programming Workshops. Lecture Notes in Business Information Processing, pp. 141–145 Springer International Publishing, Cham(2020). https://doi.org/10.1007/978-3-030-58858-8-15

---

21. Marin, E., Perino, D., Di Pietro, R.:Serverless computing: A security perspective.Journal of Cloud Computing vol.**11**(1), p 69 (2022). https://doi.org/10.1186/s13677-022-00347-w

---

22. W3C WebAssembly Working Group. https://www.w3.org/wasm/ Accessed 11 Sept 2022

23. WebAssembly Language Support Matrix. https://www.fermyon.com Accessed 29 Oct 2022

24. Haas, A., Rossberg, A., Schuff, D.L., Titzer, B.L., Holman, M., Gohman, D., Wagner, L., Zakai, A., Bastien, J.: Bringing the web up to speed with WebAssembly.In: Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 185–200 Barcelona Spain ACM (2017). https://doi.org/10.1145/3062341.3062363

25. Wang, W.: Empowering web applications with WebAssembly: Are We There Yet?In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 1301–1305. https://doi.org/10.1109/ASE51524.2021.9678831

26. Wang, Z., Wang, J., Wang, Z., Hu, Y.: Characterization and implication of edge WebAssembly runtimes. In: 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), pp. 71–80 (2021). https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00037

27. Wang, W.:How far we've come a characterization study of standalone WebAssembly runtimes.In: IISWC 2022, Austin, TX, USA (2022)

28. WASI Filesystem. WebAssembly. https://github.com/WebAssembly/wasi-filesystem Accessed 11 Sept 2022

29. WASI Sockets. WebAssembly. https://github.com/WebAssembly/wasi-sockets Accessed 11 Sept 2022

30. Wasi-Threads. WebAssembly (2022). https://github.com/WebAssembly/wasi-threads Accessed 11 Sept 2022

31. WebAssembly System Interface – Proposals. https://github.com/WebAssembly/WASI/blob/bac366c8aeb69cacfea6c4c04a503191bf1cede1/Proposals.md Accessed 11 Sept 2022

32. Gackstatter, P., Frangoudis, P.A., Dustdar, S.: Pushing serverless to the edge with WebAssembly runtimes.In:2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 140–149 (2022) https://doi.org/10.1109/CCGrid54584.2022.00023

33. Gadepalli, P.K., Peach, G., Cherkasova, L., Aitken, R., Parmer,G.: Challenges and opportunities for efficient serverless computing at the edge.In: 2019 38th Symposium on Reliable Distributed Systems (SRDS), pp. 261–2615(2019). https://doi.org/10.1109/SRDS47363.2019.00036

34. Component Model Design and Specification. WebAssembly (2022) https://github.com/WebAssembly/component-model Accessed 11 Sept 2022

35. Ling, W., Ma, L., Tian, C., Hu, Z.: Pigeon: A Dynamic and Efficient Serverless and FaaS Framework for Private Cloud.In: 2019 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE,Las Vegas, NV, USA pp. 1416–1421 (2019). https://doi.org/10.1109/CSCI49370.2019.00265

36. Karhula, P., Janak, J., Schulzrinne, H.:Checkpointing and migration of IoT edge functions.In: Proceedings of the 2nd International Workshop on Edge Systems, Analytics And Networking. dgeSys '19. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3301418.3313947

37. Pelle, I., Czentye, J., Doka, J., Kern, A., Gero, B.P., Sonkoly, B.:Operating latency sensitive applications on public Serverless Edge Cloud Platforms. IEEE Internet of Things Journal, 1–1 (2020). https://doi.org/10.1109/JIOT.2020.3042428

38. Elgamal, T.:Costless: Optimizing cost of serverless computing through function fusion and placement.In: 2018 IEEE/ACM Symposium on Edge Computing (SEC), IEEE. Seattle, WA, USA pp. 300–312. (2018). https://doi.org/10.1109/SEC.2018.00029

39. Gadepalli, P.K., McBride, S., Peach, G., Cherkasova, L., Parmer, G.:Sledge: A serverless-first, light-weight wasm runtime for the edge.In: Proceedings of the 21st International Middleware Conference. Middleware '20, Association for Computing Machinery New York, NY, USA pp. 265–279(2020). https://doi.org/10.1145/3423211.3425680

40. Long, J., Tai, H.-Y., Hsieh, S.-T., Yuan, M.J.: A lightweight design for serverless function-as-a-service. IEEE Software vol.**38**(1), pp. 75–80(2021). https://doi.org/10.1109/MS.2020.3028991arXiv:2010.07115 [cs] (2019)

41. Hockley, D., Williamson, C.: Benchmarking runtime scripting performance in wasmer.In: Companion of the 2022 ACM/SPEC International Conference on Performance Engineering.ICPE ’22, Association for Computing Machinery, New York, NY, USA pp. 97–104 (2022). https://doi.org/10.1145/3491204.3527477

42. Jangda, A., Powers, B., Berger, E., Guha, A.: Not so fast: Analyzing the Performance of WebAssembly vs. Native Code (2019). https://doi.org/10.5555/3358807.3358817

43. Ménétrey, J., Pasin, M., Felber, P., Schiavoni, V.: WebAssembly as a common layer for the coud-edge continuum.In:Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge, pp. 3–8 (2022). https://doi.org/10.1145/3526059.3533618

44. Stephen: Awesome WebAssembly runtimes (2022). https://github.com/appcypher/awesome-wasm-runtimes Accessed 11 Sept 2022

45. Containers/Crun. https://github.com/containers/crun Accessed 11 Sept 2022

46. “Distroless“ Container Images. GoogleContainerTools (2022). https://github.com/GoogleContainerTools/distroless Accessed 11 Sept 2022

47. Kmu-Bigdata/Serverless-Faas-Workbench (2021). https://github.com/kmu-bigdata/serverless-faas-workbench Accessed 15 Jan 2023

48. Kim, J., Lee, K.: FunctionBench: A suite of workloads for serverless cloud function service. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), pp. 502–504. IEEE, Milan, Italy (2019). https://doi.org/10.1109/CLOUD.2019.00091

49. Hound - Crates.Io: Rust package registry. https://crates.io/crates/hound Accessed 12 Jan 2023

50. Simon, A.N.: anthonynsimon/bild (2023). https://github.com/anthonynsimon/bild Accessed 15 Jan 2023

51. N-Body (Benchmarks Game). https://benchmarksgame-team.pages.debian.net/benchmarksgame/performance/nbody.html Accessed 15 Jan 2023

52. Prime Numbers - The Algorithms. https://the-algorithms.com Accessed 15 Jan 2023

53. Lok, A.: andylokandy/simsearch-rs (2023). https://github.com/andylokandy/simsearch-rs Accessed 15 Jan 2023

54. Potapov, S.: greyblake/whatlang-rs (2023). https://github.com/greyblake/whatlang-rs Accessed 15 Jan 2023

55. zip-rs/zip (2023). https://github.com/zip-rs/zip Accessed 15 Jan 2023

56. OpenFaaS - serverless functions made simple. https://www.openfaas.com/ 17 Jan 2023

57. Knative. https://knative.dev/ Accessed 17 Jan 2023

58. Kubeless. https://kubeless.io/ Accessed 17 Jan 2023

59. Supported WASM And WASI proposals - WasmEdge runtime. https://wasmedge.org/book/en/features/proposals.html. https://wasmedge.org/book/en/features/proposals.html Accessed 11 Jan 2023

60. Wasmedgec AOT compiler - WasmEdge runtime. https://wasmedge.org/book/en/cli/wasmedgec.html Accessed 14 Jan 2023

61. FreeBSD manual pages – clang - the clang, C, C++ and objective-C compiler. https://www.freebsd.org/cgi/man.cgi?query=clang++ &sektion=1 &manpath=FreeBSD+9.0-RELEASE Accessed 14 Jan 2023

Funding

This study was funded by the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, North Macedonia under the “NS“ project.

Author information

Authors and Affiliations

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Rudzer Boshkovikj 16, Skopje, 1000, North Macedonia

Vojdan Kjorveziroski & Sonja Filiposka

Contributions

Conceptualization: S.F. and V.K.; Investigation: V.K.; Methodology: S.F. and V.K.; Software: V.K.; Validation: S.F. and V.K.; Formal analysis: S.F. and V.K. Writing (original draft preparation): V.K.; Writing (review and editing): S.F and V.K. All authors have reviewed the manuscript.

Corresponding author

Correspondence to Vojdan Kjorveziroski.

Ethics declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Additional information

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sonja Filiposka contributed equally to this work.

Rights and permissions

About this article

Cite this article

Kjorveziroski, V., Filiposka, S. WebAssembly as an Enabler for Next Generation Serverless Computing. *J Grid Computing* **21**, 34 (2023). https://doi.org/10.1007/s10723-023-09669-8

Received	Accepted	Published
23 November 2022	27 April 2023	26 June 2023

DOI

https://doi.org/10.1007/s10723-023-09669-8

Keywords

Serverless computing **WebAssembly** **Function as a service** **Internet of things** **Performance evaluation** **Benchmarks**